**SkillSync Documentation**

**AI\_Validator documentation**

**Libraries required:**

1. PyPDF2
2. fastapi
3. google.generativeai

**Overview:**

1. This LLM model uses **Gemini Flash 1.5**, a free model, and we have integrated its API.
2. We read the PDF document using the **PyPDF2** library in Python.
3. The extracted PDF data is then sent to the LLM along with a dynamic prompt. The trainer provides **checkpoints** for the document in the form of key-value pairs, which are included in the same POST request to the API.
4. After validating the document, we receive errors (if any) in **JSON format**.
5. The project consists of two files: **main.py** and **helper.py**. The **main.py** file runs the server, while **helper.py** contains all the helper functions.

**helper.py**

read\_pdf(file: BytesIO): has parameter a binary input which would be a pdf from API.

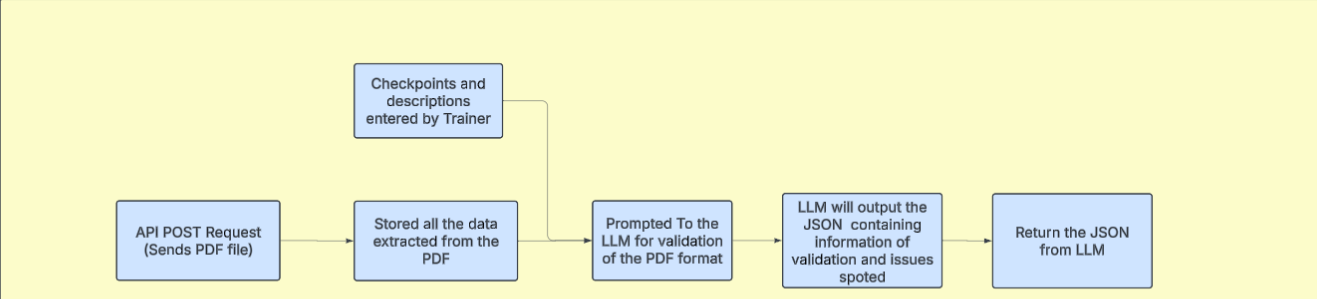
Display\_pdf\_content(pdf\_data): The argument is the pdf data extracted from previous procedure it is just to view the output its not used in main program

Generate\_prompt(pdf\_data,checkpoints): The pdf data which is extracted from the pdf is inserted in prompt and checkpoints defined by trainer are also embedded in prompt and the final prompt is ready as output of this function. This function is called in llminput() function

Llminput(pdf\_data,checkpoints): This function contains the api key and llm model api and the final prompt is hit on the LLM api to get the desired json result

**Main.py**

We have used FAST APIs, created a route @app.post("/upload-pdf/"),  
then the received JSON data which is converted to dictionary format and then llminput() function is used with parameters as pdf\_text and checkpoints\_dict.

**Flow of the pipeline:**

API doumentaion: <https://app.swaggerhub.com/apis/GeeteshChaudhari/AI_validator/1.0.0>

**AWS S3 and Azure Service Bus Integration**

OverviewThis script integrates AWS S3 for file uploads and downloads, while utilizing Azure Service Bus for message queuing and processing. It allows users to upload files, send metadata to Service Bus, process messages, store data in a database, and send verification emails.

Dependencies@aws-sdk/client-s3: AWS S3 operations

@aws-sdk/s3-request-presigner: Generates signed URLs for secure access

dotenv: Loads environment variables

@azure/service-bus: Azure Service Bus communication

express: Handles HTTP requests (assumed to be used in the project)

Environment VariablesThe script relies on the following environment variables:

S3\_BUCKET\_NAME=<your-s3-bucket-name>

CONNECTION\_STRING=<azure-service-bus-connection-string>

TOPIC\_NAME=<azure-service-bus-topic-name>

SUBSCRIPTION\_NAME=<azure-service-bus-subscription-name>Functions1.

updateMetadata(fileKey, contentType)Purpose: Updates the metadata of an S3 object if the Content-Type is missing or incorrect.

Parameters:

fileKey (string): The key (path) of the file in S3.

contentType (string): The correct MIME type.

AWS Command Used: CopyObjectCommand

2. getContentType(fileKey)Purpose: Determines the content type based on the file extension.

Returns: A MIME type string.

3. handleUpload(req, res)Purpose: Handles file uploads to AWS S3 and sends metadata to Azure Service Bus.

Steps:

Checks if the file exists in req.file.

Extracts file details (fileUrl, fileKey, name, time, batch).

Calls sendMessageToServiceBus() to queue the file details.

Sends a success response

.

4. sendMessageToServiceBus({ fileUrl, fileKey, name, time, batch })Purpose: Sends a message with file metadata to Azure Service Bus.

Steps:

Creates a Service Bus client and sender.

Sends a JSON-formatted message.

Closes the connection.

5. receiveMessages(req, res)Purpose: Listens for messages from Azure Service Bus and processes assignments.

Steps:

Subscribes to the topic.

Parses the received message.

Verifies necessary data fields.

Waits until the releaseTime.

Calls saveAssignment() to store details.

Calls sendVerificationEmails() to notify users.

Completes the message processing.

6. saveAssignment(fileKey, fileUrl, name, time, batch)Purpose: Saves assignment metadata to the database.

Database Model Used: Assignments

7. sendVerificationEmails(batch)Purpose: Sends verification emails to students of a specific batch.

Database Model Used: Student

Email Function Used: sendVerificationEmail(email)

8. handleDownload(req, res)Purpose: Generates a signed URL for downloading files from AWS S3.

Steps:

Retrieves the fileKey from request parameters.

Determines the correct Content-Type.

Checks metadata and updates if needed.

Generates a signed URL using getSignedUrl().

Returns the signed URL as a JSON response.

API EndpointsMethodEndpointDescriptionPOST/uploadUploads a file to AWS S3 and sends metadata to Azure Service Bus.GET/download/:fileKeyReturns a signed URL for file download.GET/receiveListens and processes messages from Azure Service Bus.

**SkillSync Schema and API documentation**

<https://app.swaggerhub.com/apis/GeeteshChaudhari/SkillSync/1.0.0>